CPSC 665 : An Algorithmist's Toolkit Lecture 4 : 21 Jan 2015

Linear Programming

Lecturer: Sushant Sachdeva *Scribe: Rasmus Kyng*

1. Introduction

An optimization problem requires us to find the minimum (or maximum) of a function over a domain, by finding an element of the domain on which the function is minimized (resp. maximized). The problem of minimizing a function $f : P \to \mathbb{R}$ over a domain $P \subseteq \mathbb{R}$ is usually written as

$$\min_{\boldsymbol{x} \in P} f(\boldsymbol{x}).$$

We call an optimization problem over a subset of the reals a linear program (LP) when it satisfies the following conditions

1. f is linear, i.e. $f(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x}$.
2. P is defined by half-space constraints and affine subspace constraints, i.e. "linear (in)equalities".

For example, we might define $P \subseteq \mathbb{R}$ by m inequalities

$$\begin{aligned} P = \{\boldsymbol{x} : &\boldsymbol{a}_1^T \boldsymbol{x} \le b_1 \\ &\boldsymbol{a}_2^T \boldsymbol{x} \ge b_2 \\ &\boldsymbol{a}_3^T \boldsymbol{x} = b_3 \\ &\vdots \\ &\boldsymbol{a}_m^T \boldsymbol{x} \ge b_m\} \end{aligned}$$

A constraint $\boldsymbol{a}_i^T \boldsymbol{x} = b_i$ defines an $n-1$ dimensional affine subspace, while a constraint $\boldsymbol{a}_i^T \boldsymbol{x} \ge b_i$ defines a half-space. Affine subspaces and half-spaces are both convex and because P is the intersection of the sets where each constraint is satisfied, it must also be a convex set (or empty). Dantzig introduced the definition of linear programs in 1947.

Example 1.1.

$$\begin{aligned} \max_{(x_1,x_2) \in \mathbb{R}^2} \; & x_1 + x_2 \\ \text{s.t.} \; & x_1 + x_2 \le 3 \\ & x_1 \ge 0 \\ & x_3 \ge 0 \\ & x_1 \le 2 \end{aligned}$$

The example is shown in figure 1.

We can write LPs succinctly using vector inequalities.

Definition 1.2 (Vector inequality). *For vectors $\boldsymbol{a} \in \mathbb{R}^m$ and $\boldsymbol{b} \in \mathbb{R}^m$ define $\boldsymbol{a} \ge \boldsymbol{b}$ iff $\forall i. a_i \ge b_i$.*

We say an LP is in general form when it is expressed as

$$\begin{aligned} \max_{\boldsymbol{x} \in \mathbb{R}^n} \; & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \; & A\boldsymbol{x} \ge \boldsymbol{b} \end{aligned}$$

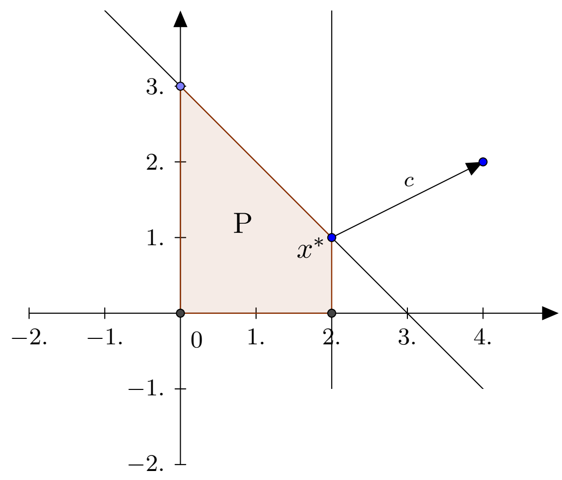

FIGURE 1. A simple example of a linear program.

An LP is said to be in standard form when it is expressed as

$$\begin{aligned} \max_{\boldsymbol{x} \in \mathbb{R}^n} & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t. } & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

HW: Every LP can be written as an equivalent LP which is in general form.
HW: Similarly, every LP can be written as an equivalent LP which is in standard form.

Definition 1.3 (**Feasible solution).** *$\boldsymbol{x}$ is feasible iff $\boldsymbol{x} \in P$.*

Definition 1.4 (**Optimal solution).** *$\boldsymbol{x}^*$ is an optimal solution iff $\boldsymbol{x}^*$ is feasible and $\boldsymbol{c}^T\boldsymbol{x}^* = \max_{\boldsymbol{x} \in P} \boldsymbol{c}^T \boldsymbol{x}$.*

Definition 1.5 (**Tight constraint).** *A constraint is tight for a point $\boldsymbol{x}$ iff it is satisfied with equality at $\boldsymbol{x}$.*

Definition 1.6 (**Basic feasible solution).** *$\boldsymbol{x}$ is a basic feasible solution iff it is feasible and has at least n tight constraints that are linearly independent.*

Constraints $\{\boldsymbol{a}_1^T\boldsymbol{x} \leq b_1, \boldsymbol{a}_2^T\boldsymbol{x} \geq b_2, \boldsymbol{a}_3^T\boldsymbol{x} = b_3, \ldots, \boldsymbol{a}_m^T\boldsymbol{x} \geq b_m\}$ are said to be linearly independent if $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_m\}$ forms a linearly independent set of vectors.

Definition 1.7 (**Unbounded LP).** *An LP $\max_{\boldsymbol{x} \in P} \boldsymbol{c}^T\boldsymbol{x}$ is said to be unbounded iff the supremum of the values it takes is $+\infty$. Similarly, a minimization program is said to be unbounded iff the infimum of its values is $-\infty$.*

Theorem 1.8. *For any LP, one of the following is true*

(1) The LP is infeasible.
(2) The LP is unbounded.
(3) The LP has a basic feasible solution $\boldsymbol{x}^*$ that is optimal.

Example 1.9. [**The Assignment Problem**]

The *Assignment Problem* requires us to match a maximal number of jobs to people that can perform them. Formally, we have a set of n jobs J, and a set of n people P, a set of allowed job assignments $E \subseteq J \times P$. We require that every job is assigned to at most one person, and that every person gets at most one job. Our goal is to find a valid job assignment $A \subseteq E$ that maximizes $|A|$.

We can express the problem as an integer program (IP). To do this, we give a bijection between an assignment A and a vector of integer variables $\boldsymbol{x} \in \{0,1\}^{|E|}$. For each $e \in E$ define $x_e \in \{0,1\}$ s.t.

$$x_e = \begin{cases} 1 & \text{if } e \in A, \\ 0 & \text{otherwise.} \end{cases}$$

The integer program is

$$\begin{aligned} \max_{\boldsymbol{x} \in \{0,1\}^{|E|}} & \sum_{e \in E} x_e \\ \text{s.t. } \forall j \in J. & \sum_{p:(j,p) \in E} x_{(j,p)} \leq 1 \\ \forall p \in P. & \sum_{j:(j,p) \in E} x_{(j,p)} \leq 1 \end{aligned}$$

Observation 1.10. OPT-IP = max assignment size. Proof: An IP solution $\boldsymbol{x}$ is feasible iff its corresponding assignment is valid, and the value of the IP is exactly the size of the corresponding assignment. Thus, if A^* is an optimal assignment, then the corresponding $\boldsymbol{x}$ achieves the same value so OPT-IP $\geq$ max assignment size. Similarly, if $\boldsymbol{x}^*$ is an optimal solution to the IP, then its corresponding assignment A has size equal to the value of the IP at $\boldsymbol{x}^*$, so max assignment size $\geq$ OPT-IP.

Remark: It is very common that combinatorial optimization problems can be expressed exactly as an IP.

We can relax the IP to an LP:

$$\begin{aligned} \max_{\boldsymbol{x} \in \mathbb{R}^{|E|}} & \sum_{e \in E} x_e \\ \text{s.t. } \forall j \in J. & \sum_{p:(j,p) \in E} x_{(j,p)} \leq 1 \\ \forall p \in P. & \sum_{j:(j,p) \in E} x_{(j,p)} \leq 1 \\ \forall e \in E. & 0 \leq x_e \leq 1 \end{aligned}$$

We call this a relaxation because the new constraints are weaker: Any feasible solution to the IP is also a feasible solution to the LP. This implies our next observation.

Observation 1.11. OPT-LP $\geq$ OPT-IP.

The assignment problem LP has an unusual property, which is captured in the next theorem.

Theorem 1.12. *For the assignment problem LP, every basic feasible solution is integral, meaning* $x_e \in \{0,1\}$ *for every* $e \in E$.

From this theorem and theorem 1.8 it follows that there exists an optimal solution to the LP which satisfies for all variables that $x_e \in \{0,1\}$, and hence we get a simple corollary.

Corollary 1.13. *OPT-IP* $\geq$ *OPT-LP.*

The main algorithms for solving general linear programs are

(1) Simplex (Dantzig, 1947). Often very fast in practice. There is no known deterministic version of the algorithm which runs in polynomial time. There is a randomized Simplex algorithm due to Kelner and Spielman that runs in polynomial time.

(2) Ellipsoid Method (Khachiyan, 1979). First polynomial time algorithm for LPs and many other convex programs. Very slow in practice.
(3) Interior Point Method (Karmarkar, 1984). Polynomial running time and often competitive with the simplex algorithm in practice.

What does it mean to solve an LP in polynomial time? By this, we mean that the running time is polynomial in n, the number of variables; m, the number of constraints; and L, the bit complexity of the problem. L can be defined in different ways, for example, the bit length of the largest number in A, $\boldsymbol{b}$, and $\boldsymbol{c}$. Even if arithmetic operations are counted as to taking constant time, all known algorithms for linear programming have a running time that depends polynomially on L.

Algorithms whose running time does not depend on L in models with constant time arithmetic, in addition to being polynomial-time under the standard bit operation model, are referred to as strongly polynomial. It is an important open problem whether there exists a strongly polynomial algorithm for linear programming.

2. Approximation Algorithm using Linear Programming

Example 2.1. [**Minimum Vertex Cover**] Given a an undirected graph $G = (V, E)$, find a subset $S \subseteq V$ which minimizes $|S|$ subject to the condition $\forall (u, v) \in E. u \in S$ or $v \in S$. The decision version of this problem is NP-complete (Karp, 1972).

There is a simple IP for this problem. To describe it, we give a bijection between a candidate vertex cover S and a vector of integer variables $\boldsymbol{x} \in \{0,1\}^{|V|}$. For each $v \in V$ define $x_v \in \{0, 1\}$ s.t.

$$x_v = \begin{cases} 1 & \text{if } v \in S, \\ 0 & \text{otherwise.} \end{cases}$$

The following IP can be shown straightforwardly to be equivalent to the minimum vertex cover problem.

$$\min_{\boldsymbol{x} \in \{0,1\}^{|V|}} \sum_{v \in V} x_v$$
$$\text{s.t. } \forall (u, v) \in E. x_u + x_v \geq 1$$

As for the assignment problem, we can relax the IP to an LP, by letting $\boldsymbol{x} \in \mathbb{R}^{|V|}$ and adding constraints $0 \leq x_v \leq 1$ for all v. Since the LP is a relaxation, we get the corollary below.

Corollary 2.2. *OPT-LP $\leq$ OPT-IP $=$ minimum vertex cover size.*

In this problem, there may be a gap between the values of the LP and the IP. For example, consider the complete graph on n vertices K_n. The minimum vertex cover for this graph has size OPT-IP $= n - 1$, while the optimal solution for the LP has $x_v = 1/2$ for all $v \in V$, and thus OPT-LP $= n/2$. In this case, as $n \to \infty$ OPT-IP/OPT-LP $= 2$.

Definition 2.3 (**Integrality Gap**)**.** *The integrality gap of an LP relaxation of an IP minimization problem is defined as*

$$\sup_{\text{instances } \mathcal{I}} \frac{\text{OPT-IP}(\mathcal{I})}{\text{OPT-LP}(\mathcal{I})}.$$

Meanwhile, the integrality gap of a relaxation of a maximization problem is defined as

$$\inf_{\text{instances } \mathcal{I}} \frac{\text{OPT-IP}(\mathcal{I})}{\text{OPT-LP}(\mathcal{I})}.$$

From the K_n example we can deduce the next corollary.

Corollary 2.4. *Integrality gap of Minimum Vertex Cover LP relaxation ≥ 2.*

Theorem 2.5. *Integrality gap of Minimum Vertex Cover LP relaxation* ≤ 2.

A common method for demonstrating an upper bound on an integrality gap is to show that a feasible solution to the relaxation can be rounded to a feasible integer solution with little or no loss in the quality of the solution.

Proof of theorem. To convert any feasible LP solution $\boldsymbol{x}$ to a feasible IP solution $\boldsymbol{z}$ we use a *rounding algorithm*, in this case a very simple one: Define the vertex set $S = \{v \in V . x_v \geq 1/2\}$. S is a vertex cover, because $\boldsymbol{x}$ feasible implies that for every edge (u, v), we have $x_u + x_v \geq 1$, and this means $\max(x_u, x_v) \geq 1/2$, so the edge is covered. Let $\boldsymbol{z}$ be the IP solution corresponding to the cover S. Now $\sum_{v \in V} z_v \leq 2 \sum_{v \in S} x_v \leq 2 \sum_{v \in V} x_v$.

In particular, by considering the optimal LP solution $\boldsymbol{x}^*$, and rounding it to a feasible IP solution $\boldsymbol{z}$, we find

$$\text{OPT-IP} \leq \sum_{v \in V} z_v \leq 2 \sum_{v \in V} x_v^* = 2 \cdot \text{OPT-LP}.$$

Definition 2.6 (Approximation Algorithm). *We say an algorithm for a minimization problem has an approximation ratio of α if for every instance $\mathcal{I}$, it outputs a solution $ALG(\mathcal{I})$ with*

$$cost(ALG(\mathcal{I})) \leq \alpha OPT.$$

For a maximization problem, we say an algorithm has an approximation ratio of α if for every instance $\mathcal{I}$, it outputs a solution $ALG(\mathcal{I})$ with

$$cost(ALG(\mathcal{I})) \geq \alpha OPT.$$

No known algorithm for Minimum Vertex Cover achieves a better approximation ratio than the $\alpha = 2$ approximation algorithm given above. Dinur and Safra showed that a 1.36 approximation algorithm for the problem would imply P = NP. Khot and Regev showed that for any constant $\epsilon > 0$, a $2 - \epsilon$ approximation algorithm would imply that the Unique Games Conjecture is false.

Example 2.7. MAX-SAT We are given n boolean variables $x_i \in \{T, F\}$ and m clauses c_j, e.g.

$$\begin{aligned} c_1 &: x_1 \vee x_2 \vee \neg x_3 \vee \ldots \\ c_2 &: x_2 \vee \neg x_4 \vee \neg x_8 \vee \ldots \\ &\vdots \\ c_m &: \neg x_1 \vee \neg x_5 \vee \neg x_6 \vee \ldots \end{aligned}$$

and we want to find some boolean assignment of our variables maximizing the number of satisfied clauses.

This is easily translated into an IP. First, we create a new integer variable $y_i \in \{0, 1\}$ for each boolean variable x_i. A bijection between the boolean variables an the integer variables is given by

$$y_i = \begin{cases} 1 & \text{if } x_i \text{ is true} \\ 0 & \text{otherwise.} \end{cases}$$

For each clause c_j we make an integer variable $z_j \in \{0, 1\}$ and add a expressing that $z_j = 1$ only if c_j is satisfied. For example, the clause c_1 in the example above gives the constraint

$$y_1 + y_2 + (1 - y_3) + \ldots \geq z_1.$$

The objective of the IP is to maximize $\sum_{j=1}^{m} z_j$, and the optimum of this IP is exactly equal to the value of the MAX-SAT problem.

We can relax this to an LP by instead allowing the y_i and z_j to range over the reals and adding constraints $0 \leq y_i \leq 1$ and $0 \leq z_j \leq 1$.

Theorem 2.8 (**Raghavan-Thomson 1987**). *Given a feasible solution y^* to the above LP, we can get a feasible solution to the corresponding IP by defining the following independent random variables*

$$x_i = \begin{cases} 1 & \text{with probability } y_i^* \\ 0 & \text{otherwise.} \end{cases}$$

Doing so gives us a solution with an expected value greater than a factor of $1 - \frac{1}{e}$ times the actual optimal value.

To prove this theorem, we first show the following lemma:

Lemma 2.9. $\Pr[c_i$ *is satisfied* $] \geq \left(1 - \frac{1}{e}\right) z_i$.

Proof of lemma. We demonstrate the methodology for the example clause c_1. The logic is easily extended to general clauses.

$$\begin{aligned} \Pr[c_i \text{ is satisfied }] &= 1 - (1 - y_1)(1 - y_2)(y_3) \\ &\geq 1 - \left(1 - \frac{y_i + y_2 + (1 - y_3)}{3}\right)^3 \\ &\geq 1 - \left(1 - \frac{z_i}{3}\right)^3 \\ &\geq \left(1 - \frac{1}{e}\right) z_i. \end{aligned}$$

In the first inequality we used the AM-GM inequality.

Proof of theorem. Denote the optimal solution of the Linear Program by OPT-LP, and the optimal solution of the MAX-SAT problem by OPT. It should be obvious that

$$\mathbb{E}[x_i] = y_i$$

Then by the linearity of expectation,

$$\begin{aligned} \mathbb{E}[\#\text{ of clauses satisfied}] &\geq \left(1 - \frac{1}{e}\right) \sum_i z_i \\ &= \left(1 - \frac{1}{e}\right) \text{OPT-LP} \\ &\geq \left(1 - \frac{1}{e}\right) \text{OPT.} \end{aligned}$$

References